Software Tools

C, Unix (Linux), and tools

Grep and regular expressions

- Grep searches for patterns in files
- Prints lines that match the pattern
- Patterns are regular expressions
- Comes (came) in a few flavours
  - Grep: uses basic regular expressions
  - Egrep: uses extended regular expressions
  - Rgrep: recursively seach directories
  - Fgrep: fixed patterns (not regular expressions)

Modern Versions of grep

- Sister commands are deprecated.
- Functionality of old sister commands is now provided by grep through flags:
  - -F: for fgrep
  - -E: for egrep
  - -r: for rgrep
- By default grep uses extended regular expressions

Regular Expressions

- Regular expressions describe a set of strings
- Have rules similar to other mathematcal expressions
- Operators can combine simpler expressions into more complex ones
- The simplest expressions match a single character
  - Almost any character matches itself
  - Bracket expressions match lists of characters enclosed in brackets “[“, “]”.
  - Bracket expressions are similar to the ones in bash.

Bracket Expressions

- The simplest is to have a list of characters between brackets
- Ranges are among the most common extensions
  - Like [a-z]
  - Can be affected by the locale
- If the first character is the carret “^” it matches the complement of the characters or ranges.
- To include the dash “-” put it at the end.

Other simple expressions

- A dot “.” matches any character
- A carret “^” matches the empty string at the beginning of a line
- A dollar “$” matches an empty string at the end of a line
- The sequences “\<” and “\>” match the empty string at the beginning and the end of a word respectively.

Repeat operators

- A star “*” matches the previous expression 0 or more times
  - Notice the difference with wildcards.
- A plus “+” matches the previous expression 1 or more times
- A question mark “?” the previous expression 0 or 1 times.
- A number in braces matches the previous expression that many times
  - Works for ranges too like {2,4} or {,6} or {2,}

Concatenation and Alternation

- Two regular expressions written one after the other match the concatenation of the respective strings
- Two regular expressions separated by a pipe “|” match either the one of the other (“OR” like)
- Alternation has lower precedence over concatenation which has lower precedence than repetition

Examples

- grep m[a-z]in\(*.c
- grep m[^a]in\(*.c
- grep m[^a].*\(*.c
- grep -E '[*+-]{2}' *.c
-

Bash Scripts

- Commands that are used interactively in bash can be put in a file and form a script.
- Bash is a sophisticated scripting language
- Scripting languages are usually interpreted and deal mainly with text
- They are intended for simple programs that integrate various software components.

Anatomy of a shell script

- Shell scripts start with #!/bin/bash
  - Awk sctipts (can) start with #!/bin/awk
  - PHP scripts start with #!/bin/php
- The comments start with pound “#” and end with a newline.
- They are usually small and serve as glue software
- Other scripting languages are awk, tcl, php, etc

Variables

- Variables work the same way as in interactive bash
- There are a few other predefined variables:
  - Name of the script: $0
  - Arguments $1, $2, ..., $9
  - Number of arguments: $#
  - All arguments: $*

Back Quotes

- To use the std output of a command as argument or assign it to a variable we use backquotes
  - Example: date=`date`; echo date is $date
- They are different from single quotes and double quotes.

Single and double quotes

- Single quotes (next to the return key) transmit the enclosed text verbatim as one argument
- Double quotes allow variable substitutions
  - Example: echo “$SHELL”; echo ‘$SHELL’